Assignment 1

(i) Program 1

a) In program i, there is an fault in the for loop. Instead of i > 0 it should be corrected to i >= 0. This causes the given test case to not properly iterate through the array and return -1 as apposed to 0.

b) The test case where the fault will not be executed is when the programs throws a NullPointerException. This means that when x is null then the program does not run and the fault is not executed.

c) The test case that executes the fault but does not result in all errors is when y=2 and x = [2, 2, 2]. This is because the expected output is 2 and even though the for loop is not iterating properly, since all the values are the same it will always find the value that is supposed to be at the end.

d) A test case that results in an error state, but not a failure is when 2 is not in the array and y=2. So for example when y=2 and x = [1, 3, 4]. This is because since it is searching for 2 and 2 is not in the array it returns -1 error.

e) The currently used test c = [2, 3, 5]; y = 2; results in -1 as apposed to 0 since it breaks out of the for loop early when i=0 and determines that 2 is not in x array.

I = 2	X = 5
I = 1	X = 3
I = 0 BREAKS LOOP	X = 2 is never checked due to error state in for loop

(ii) Program 2

a) In program ii, there is an error in the if statement. Instead of if(arr[i] <= 0) it should be if(arr[i] < 0).

b) The test case where the fault will not be executed is when the programs throws a NullPointerException. This means that when the arr is null then the program does not run and the fault is not executed.

c) The test case that executes the fault but does not result in failure is when the array does not have a 0 in it. arr = [-1, 2, 3, -4, 5] does not have a 0 and therefor the program will count accordingly and the answer will be 2.

d) There is no possibly way to give. Test case that results in an error state without failure because as long as you have no “0’s” in your array the program functions as intended. The only way that the program returns an error is if there is a 0 in the array which then provides a failure since the program will count 0 as a negative number.

e) The currently used test arr = [4, -5, 0, -5]; y results in 3 as apposed to 2 since it stays in the for loop for one iteration too long (when i=0). States are as follows:

I = 0	arr[0] = 4	counter= 0
I = 1	arr[1] = -5	counter= 1
I = 2	arr[2] = 0	counter= 2 ERROR HERE since for loop counts 0 as a negative number
I = 3	arr[3] = -5	counter= 3 Incorrect value since 0 has been counted